Sustainment of Mission Critical Electronic Warfare Software:

A Systems Engineering Approach

Ches Rehberg

WR-ALC/LNEX

Warner Robins Air Logistics Center / Electronic Combat Product Group

Purpose

- Discuss the support of embedded software using a systems engineering approach, for a critical military application domain
- Present observations from an organizations that has provided that support for over two decades

Outline

- The EW Mission and Products
- The EW Challenge:
  Continuous Change
- EW Systems Engineering
- Software Engineering in a Systems Engineering Context
- Lessons Learned

The EW Mission

Increase Aircraft Survivability

Aircraft Survivability

- Warning Functions
  - Detect and ID Radar and EO/IR Based Threat Air Defense Systems, Warn Aircrews, Cue Countermeasures
    - Radar Warning and Panoramic Receivers
    - Missile Warning Systems

Aircraft Survivability

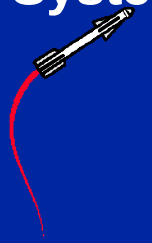
- Countermeasures Functions
  - Prevent Successful Detection, Acquisition, Tracking, and Engagement of Host Aircraft
    - RF Countermeasures
    - IR Countermeasures
    - Chaff and Flare Dispensers

The EW Product Line

Towed Decoys

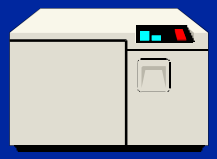
Radar Warning Systems

Missile Warning Systems

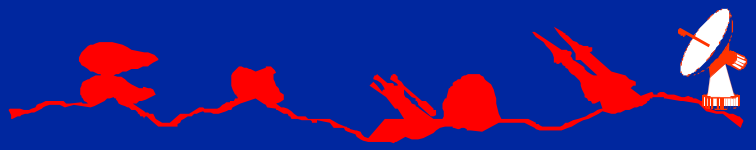


Advanced Expendables

Electronic Countermeasures Systems



Emergency & Routine Block Cycle Software Reprogramming



Electronic Countermeasures Pod

Integrated EW Suite

The EW Product Line

- Required functionality highly dependent on detailed threat characteristics:
  - RFs, pulse train details, antenna scans, other discriminants; missile and background signatures
  - Tracking and guidance receiver and control loop design
- Highly software intensive, many languages
- Complex algorithms
- Complex hardware implementations
- Infrequent major hardware upgrades

Outline

- The EW Mission and Product Line
- The EW Systems Engineering Challenge: Continuous Change
- EW Systems Engineering
- Software Engineering in a Systems Engineering Context
- Lessons Learned
-

The EW Challenge: Continuous Change

- EW system functionality requirements change drivers
  - Threat-Related Changes
    - Threat Modifications
    - Improved Knowledge of Critical Threat Characteristics
    - Countermeasures Technique Changes
    - New Threats
    - Theater-Driven Changes
  - Ops Requirements/Employment Changes
  - Integration with other On-Board Systems

The EW Challenge: Continuous Change

- **Solutions:**
  - **Acquire new EW system**
  - **Design hardware modifications, retrofit existing EW system**
  - **Change operational tactics/usage**
  - **Allocate system functional changes to software and reprogram accordingly**
    - **System software (Operational Flight Pgm)**
    - **Mission data**

Outline

- The EW Mission and Product Line
- The EW Challenge: Mission Induced Obsolescence
- EW Systems Engineering
- Software Engineering in a Systems Engineering Context
- Lessons Learned
-

EW Systems Engineering

- What is the “system”?
  - EW receiver and transmitter h/w and s/w, controls and displays
  - Threat air defense system
  - Avionics interfaces and aircraft wiring/cabling
  - Operator
  - Maintainer
  - Support equipment h/w and s/w
  - Reprogramming processes/support structure
    - System software
    - Mission data

EW Systems Engineering

- Systems engineering processes
  - Translation of operational requirements to technical requirements
  - Decomposition of requirements to successively lower levels of system
  - Test requirements development
  - Translation of requirements into design
  - Test and integration of lower level products leading to system solution
  - Project management
  - Configuration control

EW Systems Engineering

- Processes and products
  - Mission and threat analysis
  - Requirements development/translation/allocation
  - System performance characterization
  - Problem re-creation/diagnosis
  - Modeling and simulation
  - System acquisition and modification
  - Test and evaluation
  - Rapid and routine software reprogramming

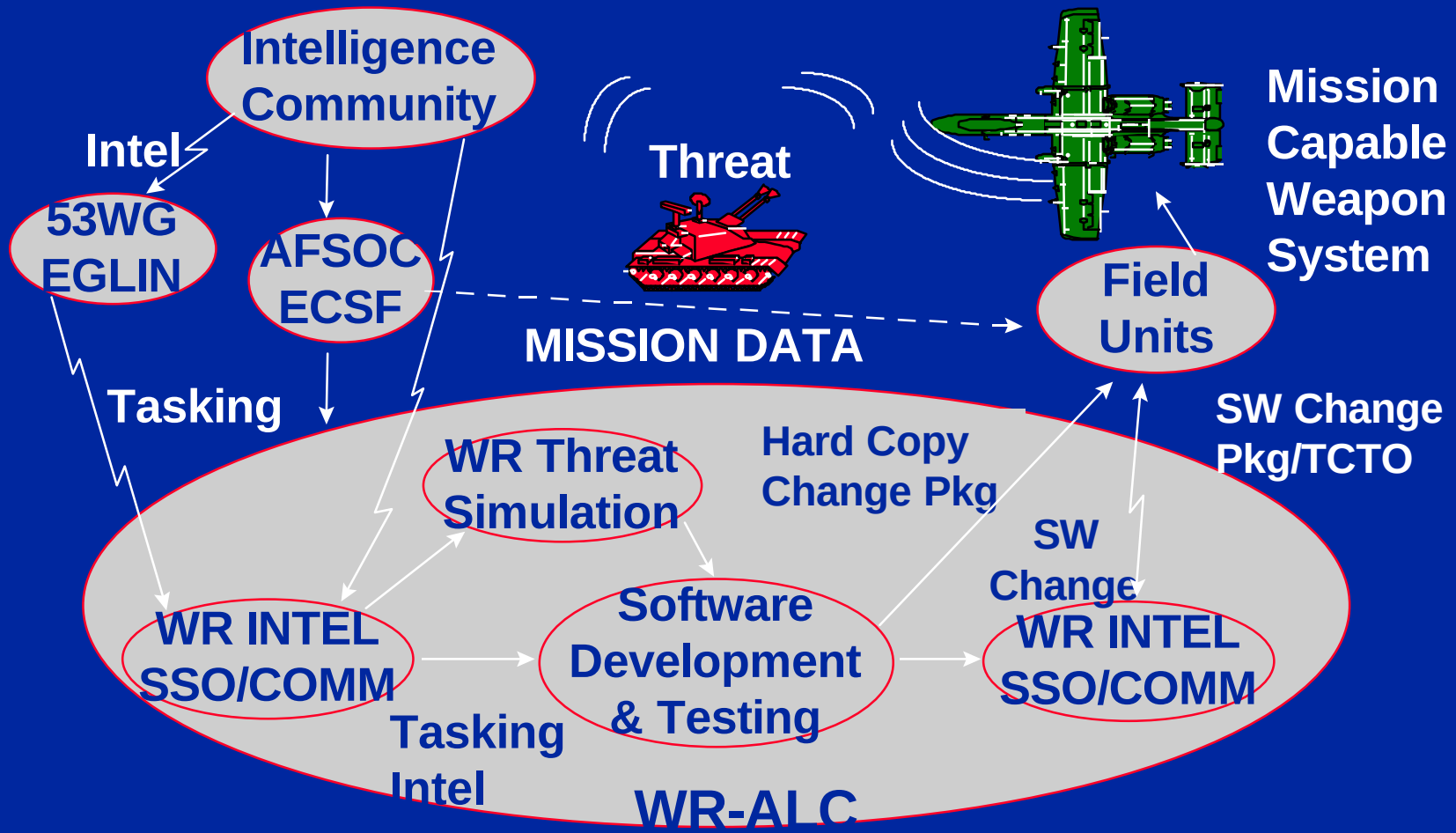
EW Systems Engineering

- **Electronic Warfare Avionics Integration Support Facility (EWAISF)**
  - Engineers and computer scientists
  - Live mockups of supported EW systems and cockpit control subsystems
  - Software support environments
  - Large scale dynamic simulations of dense threat environments at microwave frequencies
  - Avionics interface simulations
  - System and subsystem modeling tools
  - Anechoic chambers and screen rooms
  - Comm and intel support structures

Outline

- The EW Mission and Product Line
- The EW Challenge: Mission Induced Obsolescence
- EW Systems Engineering
- **Software Engineering in a Systems Engineering Context**
- Lessons Learned
-

EW Software Engineering



Emergency Reprogramming + Routine Block Cycle Updates

EW Software Engineering

- Critical process structures
  - Air Force Instruction 10-703
    - Governs entire process from intel through distribution and installation
    - Specifies emergency, urgent, routine responses
  - EC PGM Operating Instruction 10-3
    - CMM-based instruction governing all aspects of EC PGM software processes

EW Software Engineering

- Critical process activities
  - Customer requirements definition
  - System analysis
  - Allocation of rqmts to software
  - Design, code, debug, test, integration
  - Independent test
  - Customer test
  - Distribution
  - Project management/config ctrl/quality

EW Software Engineering

- Mission Data Reprogramming Concept
  - Allocate user reprogrammable tables in software
  - Provide user an interactive mission data tool
  - User reprograms system as needed
    - Simple numerical threat parameters
    - New threats
    - Change in threat priorities

Outline

- The EW Mission and Product Line
- The EW Systems Engineering Challenge: Mission Induced Obsolescence
- EW Systems Engineering
- Software Engineering in a Systems Engineering Context
- Lessons Learned

Lesson 1

- **Software support of a system built around an embedded computer is a systems engineering task**
  - **System requirements determination**
    - **Allocation of functions to software**
      - **Design/code/test of software**
  - **Integration of software in the system**
  - **System level test of modified software**

Lesson 2

Software
will
change!

Lesson 3

- System reprogrammabiilty must be addressed during the design phase
  - Allocation of desired functions to software
  - Partitioning of algorithms and data tables
  - Accessible electrical interfaces
  - User data/mission data reprogramming tools
  - File distribution methods
  - Processes and procedures

Lesson 4

100%
memory growth
requirements
aren’t enough!

Lesson 5

- **Process models and procedures are essential to success**
  - **Documented rqmts among ops customer, acquirer, supplier**
  - **Detailed plan before starting work**
    - **Work breakdown structure w/earned value**
    - **Software development plan**
  - **Detailed mid-management visibility of plans and status at least monthly**
    - **Technical status**
    - **Schedule/cost status per earned value**
    - **Risk management**

Lesson 6

Models, processes,
and structure
are essential for success
but are not substitutes
for domain knowledge.

Lesson 7

- **Modeling and simulation are essential**
  - **Requirements determination**
  - **Debug/problem re-creation**
  - **Cost savings vs. open air test**
  - **Test repeatability**